UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/021,260	12/19/2001	Carsten Driesner	30014200.1018	7197

58328 7590 07/18/2006

SONNENSCHEIN NATH & ROSENTHAL LLP
FOR SUN MICROSYSTEMS
P.O. BOX 061080
WACKER DRIVE STATION, SEARS TOWER
CHICAGO, IL 60606-1080

EXAMINER	
PRICE, NATHAN E	

ART UNIT	PAPER NUMBER
2194	

DATE MAILED: 07/18/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/021,260	DRIESNER ET AL.	
	Examiner	Art Unit	
	Nathan Price	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 April 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☒ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 35 are pending. Claims 24 – 33 have been amended.

Priority

2. Applicant's response to previous Office Action states that a certified copy of the foreign priority application will be submitted in a separate communication. The applicants are reminded that the certified copy of the foreign priority application has not been received.

Response to Arguments

3. Applicants' arguments with respect to claims 1 – 35 have been considered but are moot in view of the new ground(s) of rejection. This is partially the result of not all of the pages cited in Coulouris were provided in the last Office Action and additional pages are being cited. Further explanation is given in the rejections in response to Applicant's arguments.

Claim Rejections - 35 USC § 103

4. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

5. Claims 1 –14 and 20 – 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coulouris et al. (Coulouris, George, Jean Dollimore and Tim Kindberg. Distributed Systems Concepts and Design. Second Edition, Addison-Wesley; 1994.), hereinafter Coulouris, in view of Fidge (Fidge, Colin. "Logical Time in Distributed Computing Systems," Computer, Volume 24, Issue 8, August 1991, pages 28 – 33, ISSN: 0018-9162; retrieved from IEEE.).

As to claim 1, Coulouris discloses a method in a data processing system for synchronizing calls at a client in a server and client system, comprising the steps of:

receiving from the server a plurality of service calls generated by a plurality of threads executed at the server (page 326 ¶ 3 and page 135 ¶ 6 – 7; also: page 150 ¶ 1 – page 151 ¶ 5; page 12 ¶ 1);

receiving a synchronization call from the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change (page 326 ¶ 3, the count of events indicates the number of calls; also: page 150 ¶ 1 – page 151 ¶ 5, when a reply is required, the buffered calls are all sent and the receiver must be able to distinguish between the calls which means that the number of calls is inherently indicated. When a reply is required, the calling thread blocks: page 105 ¶ 2 – 4 and page 150 ¶ 1, 3. Previously cited portions of the text teach ordering and consistency, which also applies to newly cited portions: page 152 ¶ 2); and

placing at least one of said service calls associated with said synchronization call into a wait position, when said number of service calls indicated in said synchronization call and said number of service calls executed at the client prior to receiving said synchronization call differ (page 342 ¶ 6; also: page 152 ¶ 2, 3 and page 398 ¶ 6 and page 396 ¶ 5, the timestamps can be based on logical clocks, which effectively counts events such as requests, to maintain ordering of events and requests).

6. As to claim 1, Fidge also discloses a method in a data processing system for synchronizing calls at a client in a server and client system, comprising the steps of:

receiving from the server a plurality of service calls generated by a plurality of threads executed at the server (Fig. 1, page 29 ¶ 5 and page 30 Rule H, the processes perform events, including communication actions);

receiving a synchronization call from the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change (page 30 Rules B and F, the "ticks" count events, such as messages. When the threads block in rule F, the threads have changed and exchange the logical time, which indicates the number of calls); and

placing at least one of said service calls associated with said synchronization call into a wait position, when said number of service calls indicated in said synchronization call and said number of service calls executed at the client prior to receiving said

synchronization call differ (page 30 col. 3 ¶ 7 – 9; page 33 ¶ 3 – 4; requests are processed in the same order that they were made, not received).

7. Although Coulouris and Fidge fail to specifically disclose that synchronization is done when a thread changes, Coulouris discloses thread switching (page 173 ¶ 5) and the synchronization counts events and sends the count to the proper locations (page 326 ¶ 3). Also, Coulouris discloses buffering and sending when a thread blocks (changes). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to use vector timestamps because regardless of whether the processes run on separate machines or on the same machine, this method provides a method of synchronizing calls in the system. Also, Fidge discloses synchronization (page 30 Rule F). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to combine these references because both address the same problem of ordering with similar solutions, counters, timestamps and vector timestamps.

8. As to claim 2, the method according to claim 1 is rejected for the reasons above. Coulouris also discloses that said service calls are associated with said synchronization call by one of including respective identifiers into said at least one of said synchronization call and said service calls (page 135 ¶ 5, 6), and indicating one of a specific reception sequence and order of service of said service calls and said at least one synchronization call. Fidge also discloses the use of identifiers (page 30 Rule A). It

would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to combine these references because both address the same problem of ordering with similar solutions, counters, timestamps and vector timestamps.

9. As to claim 3, the method of synchronizing calls according to claim 1 is rejected for the reasons above. Coulouris also discloses that said receiving steps include receiving a first call sequence of a plurality of call sequences from the server, said first call sequence including a first synchronization call and at least one service call from a first thread, said first synchronization call including a first server call counter value indicating a first number of service calls executed at the server prior to the first synchronization call (page 342 ¶ 5; also: page 151 ¶ 5);

said method further comprising the step of:

comparing said first server call counter value with a client call counter value, said client call counter value indicating a second number of service calls executed at the client prior to receiving said first synchronization call (page 342 ¶ 6 including listed criteria; also: page 152 ¶ 2 – 3 and page 298 ¶ 6); and

one of:

executing said first number of service calls of said first call sequence and counting said executed first number of service calls using a client call counter value, if said client call counter value and said first server call counter value coincide (page 342 step 3 and ¶ 6); and

placing said first call sequence into a wait position, if said client call counter value and said first server current call counter differ (page 342 ¶ 6).

10. As to claim 4, the method according to claim 1 is rejected for the reasons above. Both Coulouris and Fidge also disclose that said service calls are generated asynchronously (Coulouris: page 152 ¶ 3; Fidge: page 30 Rules F – H). See the rejection of claim 1 for motivation to combine.

11. As to claim 5, the method according to claim 3 is rejected for the reasons above. Coulouris also discloses that the method further comprises the steps of:

determining whether a second call sequence in a wait position is available, said second call sequence including a plurality of service calls from a second thread executed at the server and a second synchronization call including a second server call counter value indicating a third number of service calls executed at the server prior to said second synchronization call (page 342 ¶ 5 – criteria list of ¶ 6);

wherein if said second call sequence in a wait position is not available, waiting to receive further service calls and synchronization calls (page 342 ¶ 6); and

wherein if said second call sequence is available, determining that said second server call counter value coincides with said client call counter value, and executing said third number of service calls of said second call sequence and incrementing said client counter value for each executed third number of service calls (page 342 ¶ 5 – criteria list of ¶ 6).

12. As to claim 6, the method according to claim 5 is rejected for the reasons above. Coulouris also discloses that the method further comprises the step of:

waiting for a third call sequence to be received from the server unit, the third call sequence including a third synchronization call including a third server call counter value coinciding with said client call counter value (page 342 ¶ 5 – criteria list of ¶ 6).

13. As to claim 7, the method according to claim 3 is rejected for the reasons above. Coulouris also discloses that said call sequences are received as groups included into packets from the server, each group being generated upon one of a timer signal at the server (page 151 ¶ 5), a synchronous call at the server, and a synchronization call at the server.

14. As to claim 8, the method according to claim 2 is rejected for the reasons above. Coulouris also discloses that said synchronization call and said service calls are received in an arbitrary order (page 325 ¶ 1).

15. As to claim 9, the method according to claim 1 is rejected for the reasons above. Coulouris also discloses that said service calls from said plurality of threads at the server are executed in corresponding threads at the client (page 135 ¶ 7).

16. As to claim 10, the method according to claim 3 is rejected for the reasons above. Coulouris also discloses that said first server call counter value indicates a total number of service calls at the server executed prior to a current service call and requires communication with the client (page 326 ¶ 3); and

wherein said client call counter value indicates a total number of service calls executed at the client and involves communication with the server (page 326 step 3).

17. As to claim 11, the method according to claim 1 is rejected for the reasons above. Coulouris also discloses that each of said service calls from the server includes at least one of:

obtaining instructions to display information on a display of the client (page 149 ¶ 10);

rendering instructions;

storing instructions to store information at the client; and

information on processing results from the server.

18. As to claim 12, Coulouris discloses a method in a data processing system for synchronizing calls at a server in a server and client system, comprising the steps of:

transmitting a plurality of service calls generated by a plurality of threads at the server to a client (page 326 ¶ 3 and page 135 ¶ 6 – 7; also: page 150 ¶ 1 – page 151 ¶ 5);

generating a synchronization call when a thread of said plurality of threads executed at the server changes, said synchronization call indicating a number of service calls generated by said plurality of threads at the server prior to the thread change (326 ¶ 3; also: page 150 ¶ 1 – page 151 ¶ 5, page 106 ¶ 2 – 4, page 152 ¶ 2); and

transmitting said synchronization call to the client to allow the client to synchronize a service call execution (page 326 step 2; also: page 150 ¶ 1 – page 151 ¶ 5).

19. As to claim 12, Fidge also discloses a method in a data processing system for synchronizing calls at a server in a server and client system, comprising the steps of:

transmitting a plurality of service calls generated by a plurality of threads at the server to a client (page 29 ¶ 5 and page 30 Rule G);

generating a synchronization call when a thread of said plurality of threads executed at the server changes, said synchronization call indicating a number of service calls generated by said plurality of threads at the server prior to the thread change (page 30 Rule B and F); and

transmitting said synchronization call to the client to allow the client to synchronize a service call execution (page 30 col. 3 ¶ 3 – 9).

20. Although Coulouris and Fidge fail to specifically disclose that synchronization is done when a thread changes, Coulouris discloses thread switching (page 173 ¶ 5) and the synchronization counts events and sends the count to the proper locations (page

326 ¶ 3). Also, Coulouris discloses buffering and sending when a thread blocks (changes). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to use vector timestamps because regardless of whether the processes run on separate machines or on the same machine, this method provides a method of synchronizing calls in the system. Also, Fidge discloses synchronization (page 30 Rule F). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to combine these references because both address the same problem of ordering with similar solutions, counters, timestamps and vector timestamps.

21. As to claim 13, the method according to claim 12 is rejected for the reasons above and the addition limitations are rejected for the same reason as those of claim 2.

22. As to claim 14, the method of according to 12 is rejected for the reasons above and the addition limitation is rejected for the same reason as that of claim 4.

23. As to claim 20, the method according to claim 12 is rejected for the reasons above and the additional limitation is rejected for the same reason as that of claim 11.

24. As to claim 21, the method according to claim 12 is rejected for the reasons above. Coulouris also discloses that a synchronization call is further generated upon an occurrence of one of the group comprising:

a timer signal (page 151 ¶ 5);

a predetermined number of service calls; and

a synchronous call.

25. As to claim 22, Coulouris discloses a method in a data processing system for synchronizing calls in a client and server system, the method comprising the steps of:

transmitting a plurality of service calls generated by a plurality of threads executed at the server to the client (page 326 ¶ 3; also: page 150 ¶ 1 – page 151 ¶ 5);

generating a synchronization call at the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change (page 326 ¶ 3; also: page 150 ¶ 1 – page 151 ¶ 5, page 106 ¶ 2 – 4, page 152 ¶ 2);

transmitting said synchronization call to the client to allow the client to synchronize a service call execution (page 326 step 2; also: page 150 ¶ 1 – page 151 ¶ 5);

receiving said synchronization call at the client (page 326 steps 2 – 3); and

placing at least one of said service calls associated with said synchronization call into a wait position, if said number indicated in said synchronization call and said number of service calls executed at the client prior to receiving said synchronization call differ (page 342 ¶ 6; also: page 152 ¶ 2, 3, page 398 ¶ 6 and page 396 ¶ 5).

26. As to claim 22, Fidge also discloses a method in a data processing system for synchronizing calls in a client and server system, the method comprising the steps of:

transmitting a plurality of service calls generated by a plurality of threads executed at the server to the client (page 29 ¶ 5 and page 30 Rule G);

generating a synchronization call at the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change (page 30 Rules B and F);

transmitting said synchronization call to the client to allow the client to synchronize a service call execution (page 30 Rules G – I);

receiving said synchronization call at the client (page 30 Rule H); and

placing at least one of said service calls associated with said synchronization call into a wait position, if said number indicated in said synchronization call and said number of service calls executed at the client prior to receiving said synchronization call differ (page 30 Rules H – Comparison Property).

27. Although Coulouris and Fidge fail to specifically disclose that synchronization is done when a thread changes, Coulouris discloses thread switching (page 173 ¶ 5) and the synchronization counts events and sends the count to the proper locations (page 326 ¶ 3). Also, Coulouris discloses buffering and sending when a thread blocks (changes). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to use vector timestamps because regardless of whether the

processes run on separate machines or on the same machine, this method provides a method of synchronizing calls in the system. Also, Fidge discloses synchronization (page 30 Rule F). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to combine these references because both address the same problem of ordering with similar solutions, counters, timestamps and vector timestamps.

28. As to claim 23, in order to implement the disclosure made by Coulouris and Fidge, including an implementation of the method of claim 22, one of ordinary skill in the art would make use of a computer readable medium containing instructions that cause a data processing system to perform the method of claim 22. Therefore, claim 23 is rejected for the same reasons as the rejection of claim 22.

29. As to claim 24, the computer readable medium of claim 23 is rejected for the reasons above. The limitation added by claim 24 is rejected for the same reasons as the limitation added by claim 2.

30. As to claims 25, 26, 31 and 33, the method according to claim 24 is rejected for the reasons above. The additional limitations of claims 25, 26, 31 and 33 are rejected for the same reasons as the limitations added by dependent claims 3, 4, 9 and 11, respectively.

31. As to claims 27, 29, 30 and 32, the method according to claim 26 is rejected for the reasons above. The additional limitations of claims 27, 29, 30 and 32 are rejected for the same reasons as the limitations added by dependent claims 3, 4, 9 and 11, respectively.

32. As to claim 28, the computer readable medium of claim 27 is rejected for the reasons above. The limitation added by claim 28 is rejected for the same reasons as the limitation added by claim 6.

33. As to claim 34, Coulouris discloses a data processing system for synchronizing calls in a client and server system, the data processing system comprising:

a client computer comprising (Figure 1.1):

a memory including a client program that receives a plurality of service calls generated by a plurality of threads executed at the server, that receives a synchronization call from the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change (page 326 ¶ 3), and that places at least one of said service calls associated with said synchronization call into a wait position, if said number indicated in said synchronization call and said number of service calls executed at the client prior to receiving said synchronization call differ (page 342 ¶ 6; also: page 150 ¶ 1 – page 151 ¶ 5, page 152 ¶ 2, 3, page 396 ¶ 5, page 398 ¶ 6 and page 105 ¶ 2 – 4); and

a processor that runs said client program (inherent);

a server computer comprising (Figure 1.1):

a memory including a server program that transmits a plurality of service calls generated by a plurality of threads at the server to the client, that generates a synchronization call when a thread of said plurality of threads executed at the server changes, said synchronization call indicating a number of service calls generated by said plurality of threads at the server prior to the thread change, and that transmits said synchronization call to the client to allow the client to synchronize a service call execution (page 326 ¶ 3 and step 2; also: page 150 ¶ 1 – page 151 ¶ 5, page 152 ¶ 2, 3, page 396 ¶ 5, page 398 ¶ 6 and page 105 ¶ 2 – 4); and

a processor that runs said server program (inherent); and

a network connecting said client computer and said server computer (Figure 1.1).

34. In order to implement the distributed computing system disclosed by Fidge (see preceding rejections), a computer system with the various computer components and requirements are inherently required.

35. Although Coulouris and Fidge fail to specifically disclose that synchronization is done when a thread changes, Coulouris discloses thread switching (page 173 ¶ 5) and the synchronization counts events and sends the count to the proper locations (page 326 ¶ 3). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to use vector timestamps because regardless of whether the

processes run on separate machines or on the same machine, this method provides a method of synchronizing calls in the system. Also, Fidge discloses synchronization (page 30 Rule F). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to combine these references because both address the same problem of ordering with similar solutions, counters, timestamps and vector timestamps.

36. As to claim 35, Coulouris discloses an apparatus for synchronizing calls in a client and server system, the apparatus comprising:

means for transmitting a plurality of service calls generated by a plurality of threads executed at the server to the client (page 326 ¶ 3);

means for generating a synchronization call at the server, said synchronization call indicating that one of said plurality of threads executed a the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change (page 326 ¶ 3);

means for transmitting said synchronization call to the client to allow the client to synchronize a service call execution (page 326 step 2);

means for receiving said synchronization call at the client (page 326 steps 2 – 3); and

means for placing at least one of said service calls associated with said synchronization call into a wait position, if said number indicated in said synchronization call and said number of service calls executed at the client prior to receiving said

synchronization call differ (page 342 ¶ 6). See also: page 150 ¶ 1 – page 151 ¶ 5, page 152 ¶ 2, 3, page 396 ¶ 5, page 398 ¶ 6 and page 105 ¶ 2 – 4.

37. In order to implement the system disclosed by Fidge (see preceding rejections), the means for performing the disclosed steps are inherently required.

38. Although Coulouris and Fidge fail to specifically disclose that synchronization is done when a thread changes, Coulouris discloses thread switching (page 173 ¶ 5) and the synchronization counts events and sends the count to the proper locations (page 326 ¶ 3). Also, Coulouris discloses buffering and sending when a thread blocks (changes). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to use vector timestamps because regardless of whether the processes run on separate machines or on the same machine, this method provides a method of synchronizing calls in the system. Also, Fidge discloses synchronization (page 30 Rule F). It would have been obvious to one of ordinary skill in the art at the time of the applicants' invention to combine these references because both address the same problem of ordering with similar solutions, counters, timestamps and vector timestamps.

39. Claims 15 – 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coulouris in view of Fidge as applied to claim 12 above, and further in view of Liedtke (Liedtke, Jochen. "Improving IPC by Kernel Design," ACM Symposium on Operating

Systems Principles, Proceedings of the fourteenth ACM Symposium on Operating Systems Principles, ACM Press, 1994; pages 175 – 188.).

40. As to claim 15, Coulouris discloses the step of:

generating a current service call by a first thread executed at the server (page 326 ¶ 3);

generating a first synchronization call including a server call counter value indicating a number of service calls executed at the server prior to said current service call and transmitting said first synchronization call to the client (page 326 ¶ 3 and step 2), for enabling the client to synchronize an execution of a plurality of service calls from at least said first thread and said second thread (page 326 – 327 vector clock update algorithm); and

counting said current service call using said server call counter value (page 326 ¶ 3). See also: page 150 ¶ 1 – page 151 ¶ 5, page 152 ¶ 2, 3, page 396 ¶ 5, page 398 ¶ 6 and page 105 ¶ 2 – 4.

41. Coulouris fails to specifically disclose determining and comparing thread identifiers and carrying out the appropriate action depending on whether or not the identifiers differ.

42. Liedtke discloses using unique identifiers to distinguish between threads (page 180 section 5.3.1). It would have been obvious to one of ordinary skill in the art at the

time of the applicants' invention to use the thread IDs to determine if two calls were made by the same thread or different threads in order to confirm a thread change because the thread IDs are unique which provides a way to distinguish between threads. Comparing thread IDs provides confirmation that a thread change occurred, regardless of whether or not a change was requested or expected. The appropriate response can then be performed. It would have been obvious to combine these disclosures because Coulouris discloses the use of multiple processes and the Liedtke discloses identifiers that can be used to distinguish between the processes.

43. As to claim 16, the method according to claim 15 is rejected for the reasons above. The limitation added by claim 16 is rejected for the same reasons as the limitation added by claim 7.

44. As to claim 17, the method according to claim 15 is rejected for the reasons above. Coulouris also discloses that said synchronization call includes said second thread identifier of said second thread, and said number of service calls include a thread identifier of each thread generating said service call (page 326 ¶ 3, the vector timestamp provides information regarding the other processes); and

Wherein said synchronization call and said number of service calls are transmitted to the client in an arbitrary order (page 325 ¶ 1).

45. As to claim 18, the method according to claim 15 is rejected for the reasons above. See claim 9 for the rejection of the additional limitation of claim 18.

46. As to claim 19, the method according to claim 15 is rejected for the reasons above. Coulouris also discloses that said server call counter value indicates a total number of service calls requiring communication with the client executed at the server, prior to the current service call (page 326 ¶ 3). See also: page 396 ¶ 5 and page 398 ¶ 6.

Conclusion

47. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nathan Price whose telephone number is (571) 272-4196. The examiner can normally be reached on 7:30am - 4:00pm, Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on (571) 272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

NP

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER